Exhibit C

```perl
#!/u/bmaison/bin/perl5

$rulefile=$ARGV[0];

$filein=$ARGV[1];
#"twov03_new4.data";
#$filein="twov114_new4.test";

$fileout=$ARGV[2];


open(f,$filein) || die "I can't";

$count=0;

$thresh_rules=1;

$print = 1;

$count=0;
while(<f>){
  chop;
  $sentence[$count]=$_;
  @a=split(/,/);
  $truth[$count]=$a[12];
  if ($#a > 12){
    $prediction[$count]=$a[$#a];
  }
  else{
    $prediction[$count]=1;
  }
  for ($i=0; $i<=$#a; $i++){
    $sent[$count][$i] = $a[$i];
  }
  $count++;
}
$no_sent=$count;


close(f);

print "HERE\n";

open(frules,$rulefile) || " I can't open $rulefile";
while(<frules>){
  chop;

  $good=0;
```

```perl
$bad=0;
$changed=0;

@a=split(/\s+/);
  print "RULE $_ \t";
  $rule = $_;
  @b=split(/:/,$a[0]);
  $source=$b[0];
  $target=$b[1];

  @d=split(/,/,$a[1]);
  for ($i=0; $i<=$#d; $i++){
    @c=split(/:/,$d[$i]);
    $attribute[$i]=$c[0];
    $comparison[$i]=$c[1];
    $value[$i]=$c[2];
    #print $attribute[$i],"\t", $comparison[$i],"\t",$value[$i],
"\n";
  }

  for ($i=0; $i< $no_sent; $i++){
    if ($prediction[$i] eq $source){
      $count=0;
      for ($j=0; $j<=$#d; $j++){
        if ($comparison[$j] eq "eq"){
          if ($sent[$i][$attribute[$j]] ne $value[$j]){
            last;
          }
          else{
            $count++;
          }
        }
        elsif($comparison[$j] eq "ls"){
          if ($sent[$i][$attribute[$j]] > $value[$j]){
            last;
          }
          else{
            $count++;
          }
        }
        elsif($comparison[$j] eq "gt"){
          if ($sent[$i][$attribute[$j]] < $value[$j]){
            last;
          }
          else{
            $count++;
          }
        }
```

```perl
      else{
         print "ERROR not such a comparison $comparison\n";
         exit;
      }
   }
   #print $i,"\t",$count,"\t",$#d+1,"\n";
   if ($count == $#d+1){
      $changed++;
      if ($prediction[$i] eq $truth[$i]){
         $bad++;
      }
      else{
         $good++;
      }
      $prediction[$i] = $target;
      #print "I changed $source to $target for $i\n";
   }
   }
   }
   $diff=$good-$bad;
   if ($diff > $maxdiff){
      $maxdiff=$diff;
      $bestrule = $rule;
   }
   print "  GOOD:$good  BAD:$bad  ALL:$diff\n";
}

$count_errors=0;
$count1_2=0;
$count2_1=0;

for ($i=0; $i< $no_sent; $i++){
   if ($prediction[$i] ne $truth[$i]){
      $count_errors++;
      if ($prediction[$i] eq "1" &&  $truth[$i] eq "2"){
         $count2_1++;
      }
      elsif($prediction[$i] eq "2" &&  $truth[$i] eq "1"){
         $count1_2++;
      }
   }
   }
}

print "MAXDIFF $maxdiff  $bestrule\n";
print "ALL ERRORS  $count_errors\n";
print "TRUTH = 1 PREDICTION = 2 :  $count1_2\n";
print "TRUTH = 2 PREDICTION = 1 :  $count2_1\n";
```

```perl
if ($print){
  open(fout,">$fileout");
  for ($i=0; $i< $no_sent; $i++){
    print fout $sentence[$i],",",$prediction[$i],"\n";
  }
}
```